Logical Traps in High-Level Knowledge and Information Fusion

Éric Grégoire
CRIL CNRS & IRCICA
Université d'Artois
Rue Jean Souvraz SP18
F-62307 Lens Cedex
FRANCE

gregoire@cril.univ-artois.fr

ABSTRACT

In this paper, we are concerned with high-level information fusion, such as it is envisioned in JDL levels 4 and 5. More precisely, the focus is on decision and reasoning systems that must act in a rational and logical way from several knowledge and high-level information sources and databases. Whatever the selected knowledge and information representation languages in the sources are, an automatic reasoning system that is part of decision-support system must obey unquestionable rules and principles pertaining to logic. However, applying those rules that are initially defined for reasoning about a unique view of a situation, which is supposed to be fully described, can lead to serious and unexpected drawbacks in the context of the fusion of several different and partial views. We address three such issues. The first-one is related to the dramatic trivialising effects of logical inconsistency. The second one lies on the implicit transformation of necessary conditions into sufficient ones when fusion is operated. The last one is the implicit loss of more specific and precise information in favour of more general knowledge. Each of these problems is motivated by intuitive examples before practical solutions are discussed.

1. INTRODUCTION

In this paper, the focus is on information fusion when the fused information is actually high-level knowledge, such as decision rules. We are thus concerned with high-level information fusion, such as it is envisioned in JDL levels 4 and 5. Complex and advanced decision-support systems that are based on the fusion of such knowledge coming from various sources and databases, are expected to act in a rational and logical way. Whatever the selected knowledge and information representation languages in the sources are, they must obey unquestionable rules and principles pertaining to logic. However, providing them with fully standard logical deductive abilities can lead to severe drawbacks. In this paper, we concentrate on three different problems that can occur due to the fusion process itself. The first issue is related to the dramatic trivialising effects of logical inconsistency. The second one lies on the implicit transformation of necessary conditions into sufficient ones when fusion is operated. The last one is the implicit loss of more specific and precise information in favour of more general knowledge. Each of these problems is motivated by intuitive examples before practical solutions are discussed.

2. TRIVIALISING EFFECTS OF LOGICAL INCONSISTENCY

Complex and advanced decision-support systems must often rely on the fusion of several different information sources and databases that can convey elaborate forms of knowledge, such as complex decision rules. Obviously enough, we normally assume that such systems will act in a rational way.

Grégoire, É. (2006) Logical Traps in High-Level Knowledge and Information Fusion. In *Information Fusion for Command Support* (pp. 7-1 – 7-10). Meeting Proceedings RTO-MP-IST-055, Paper 7. Neuilly-sur-Seine, France: RTO. Available from: http://www.rto.nato.int/abstracts.asp.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **01 DEC 2006**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**
4. TITLE AND SUBTITLE **Logical Traps in High-Level Knowledge and Information Fusion**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **CRIL CNRS & IRCICA Université dArtois Rue Jean Souvraz SP18 F-62307 Lens Cedex FRANCE**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release, distribution unlimited**		
13. SUPPLEMENTARY NOTES **See also ADM002031., The original document contains color images.**		
14. ABSTRACT		
15. SUBJECT TERMS		

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**UU**	**34**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Since it is based on pieces of information that are not just mere numerical data but can be high-level patterns of knowledge and reasoning, we thus naturally assume that uncontroversial rules of logic will apply. For instance, if we have both that "if *A* then conclude *B*" and "*A* is true", then it should be able to conclude *B*. Or, if we have both that "Either *A* or *B* is true" and that "A" is false, then it should be able to conclude that "B" is true. Accordingly, it would be tempting to attempt to provide these systems with fully complete and sound logical reasoning abilities, at least modulo computational limits. However, a perfect (standard) logical reasoner can exhibit a behaviour that leads to undesirable conclusions. One such a problem occurs when the involved knowledge exhibits some contradiction.

For example, assume that a first source of information contains *sensor1_ ok* whereas a second one asserts the opposite view, namely *not(sensor1_ok).* When we assume that the reasoning system is perfectly rational and obeys all usual unquestionable logical rules, such a single contradiction within a possibly very large amount of information has a dramatic effect on the actual deductive capacity of the reasoning system. Indeed, *any* conclusion and its contrary can be deduced from a single logical contradiction, when a sound and complete deductive system is considered. From the example, a perfect rational deductive artificial (or real) agent could deduce *any* conclusion at the same time: e.g. *must_launch_missile* and *not(must_launch_missile)*.

This is thus a serious issue and several approaches to handle it are the object of many research efforts. Mainly, we can distinguish between two families of approaches to resolve such a drawback.

On the one hand, logical systems can be weakened to avoid the systematic propagation of inconsistency. Specific logical systems, like paraconsistent logics, have been devised to that end. Another approach involves reasoning systems that deal with maximal consistent subsets of logical consequences. The idea is to reason from maximal subsets of the information that remain contradiction-free.

On the other hand, we can keep the full logical deductive apparatus but apply it after a first check that the fused knowledge does not involve any contradiction has been performed. In this respect, it should be noted that checking for logical contradictions is a heavy computational task, even in basic languages like the Boolean framework. Indeed, checking the satisfiability of a set of propositional clauses is a NP-complete problem. Fortunately, recent research progress shows that it is tractable for many instances.

3. NECESSARY CONDITIONS CAN BECOME SUFFICIENT ONES

3.1. Motivating example

Assume that a decision-support system has to fuse two different knowledge sources. The first-one expresses information about the technical capacity of a defence system, like "*If a target is detected and is not out of reach then open fire against the target*". The second one is an opportunity and targets selection module asserting e.g. "*If a target is detected and if there is not a more seriously threatening target then open fire against the target*". If the two sources are not contradicting one another, then usual logic-based approaches to knowledge fusion will simply take the set-theoretic union of the involved formulas. As a consequence, the two conditions for opening fire, which were to be interpreted as necessary ones in the initial sources will be considered as sufficient ones in the fused knowledge. Accordingly, if a target is indeed detected and is not out of reach, then fire is opened against the target even if there exists a more seriously threatening one. Clearly, this is a wrong decision and the resulted fused knowledge should be "*If a target is detected and is not out of reach and there is not a more seriously threatening target then open fire against the target*".

Circumscribing this problem in the above example is straightforward since it can be achieved by fusing the involved rules themselves. However, finding necessary conditions that can be wrongly interpreted as

sufficient ones when fusion is operated is not easy in the general case. In [10], we discuss a polynomial-time pre-processing step on the knowledge sources that prevents such a problem from occurring, at least in a specific but standard knowledge representation language. Let us briefly expose this.

3.2. A standard knowledge representation framework

We consider a propositional language **L** of formulas over a finite alphabet **P** of Boolean variables, also called *atoms* or *propositions*. The $\wedge$, $\vee$, $\neg$ and $\Rightarrow$ symbols represent the standard conjunctive, disjunctive, negation and material implication connectives, respectively. A *literal* is an atom or a negated atom. From a syntactical point of view, a *knowledge base KB* will be a multiset of formulas of **L**. Actually, we shall often assume that these formulas are in CNF (conjunctive normal form), i.e. are a conjunction of clauses, where a clause is a disjunction of literals. In the following, we consider a multi-set of $n > 1$ propositional knowledge bases E = $\{KB_1,\ldots,KB_n\}$ to be fused. **Ab** is a subset of **P** containing Boolean variables noted $Ab_1,\ldots, Ab_m$, where Ab_i ($i \in [1..m]$) propositions are called abnormality propositions and are intended to represent unexpected faulty behaviors [16] [17]. For example, the rule "When the switch is on and when the switch is not out of service, then the lights are on" is represented by the formulas.

$Switch\_on \wedge \neg Ab_1 \Rightarrow Lights\_on$

$Out\_of\_service\_switch \Rightarrow Ab_1$

In the description of a physical device, we require the knowledge engineer to introduce abnormality propositions in most rules, in order to allow the representation of -unexpected- failures. We shall assume that each KB_i will use a different subset of **Ab**, and that these subsets share an empty intersection. An alternative approach would consist in linking each physical component to a given abnormality proposition, allowing to mark its faulty status. But we believe that this would lead to a loss of expressiveness because all sources of possible failures would be connected to a same abnormality proposition.

In this framework, Ab_i propositions are thus given a quite limited epistemological role. They are not intended to encode every form of exceptions to rules as in McCarthy's proposal [16], as required in complex taxonomic representation schemata like e.g. inheritance nets, or in complex forms of non-monotonic reasoning. Actually, they are just intended to represent sufficient conditions for unexpected faulty behavior, only. Obviously enough, it would be possible to incorporate other Ab_i with other roles in another set, different from **Ab**, and provide them with an adequate treatment.

Such an approach to represent technical specification is also part of the standard framework of the so-called consistency-based approaches to diagnosis [14], following the seminal work by Reiter [17]. In this domain, the deep -i.e. from first principles- knowledge of a physical device or process is described using the above language (in a first-order setting). Together with some observations (facts, mainly) and a list of components, a troublesome behavior of the device is translated by logical inconsistency of the whole set of formulas, under the assumption that all abnormality propositions are *false*. Diagnosis amounts to detecting a set of abnormality propositions that would restore consistency when assigned *true*, this set being most often required to be minimal w.r.t. cardinality. *Let us stress that this paper is not intended to be a direct contribution to this very close domain of research.* The assembly of knowledge components that is considered here is not necessarily devoted to diagnosis checking. Moreover, it does not necessarily involve knowledge from first principles and is studied at the creation phase of the technical specification knowledge, not necessarily under an operation mode of the device or the process. Most importantly, we shall, among other things, consider the situation where the assembled knowledge is inconsistent, independently of working observations, and even when no abnormality propositions can restore consistency by setting them to *true*.

Finally, let us recall the basic semantical definitions we shall need in the following. Let Ω denote the set of all *interpretations* of **L**, which are functions assigning either *true* or *false* to every atom. A *model* ω of *KB* is an interpretation of Ω that satisfies every formula of *KB*. An interpretation or a model will be represented by the set of literals that it satisfies. The set of models of *KB* will be denoted [[*KB*]]. *KB* is *consistent* when [[*KB*]] is not empty. $KB \models f$ expresses that the formula f can be deduced from *KB*, i.e. that it is *true* in all models of *KB*. We opt for a semantical (vs. a purely syntactical) regard of a *KB*. Under this point of view, a *KB* is thus the set of formulas that it contains together with its deductive consequences. Actually, the models that correctly describe a *KB* are models that minimize the number of abnormality propositions set to *true*. Indeed, failures are expected not to occur. From a proof-theoretical point of view, roughly this amounts to the circumscription of the set of formulas with respect to abnormality propositions [16].

3.3. Which formulas should be merged?

In the following, we thus consider a multi-set of $n > 1$ propositional knowledge-bases $E = \{KB_1,\ldots,KB_n\}$ to be fused. As explained above, we need to merge *some types* of formulas themselves if we want to correctly represent the assertion of multiple necessary conditions for the proper functioning of a physical device or process. In this section, we investigate the various possible combinations of formulas that should be merged, from an intuitive point of view. A more formal and general characterization is provided in the next section. Table 1 gives some types of couples of formulas involving abnormality propositions. Let us consider them successively and analyze whether they are to be merged or not.

In the general case, we cannot forecast all possible circumstances for bad behavior of a device. Accordingly, Ab_i propositions can only represent sufficient (vs. necessary) conditions for faulty behavior, whereas $\neg Ab_i$ literals can only represent necessary (vs. sufficient) conditions for proper functioning. As explained in the introduction, this remark justifies our decision of merging formulas of Table 1.(1) when they belong to two different *KB*s: $A \wedge \neg Ab_1 \Rightarrow B$ and $A \wedge \neg Ab_2 \Rightarrow B$ are merged to form $A \wedge \neg Ab_1 \wedge \neg Ab_2 \Rightarrow B$. Now, it might happen that both formulas $A \wedge \neg Ab_1 \Rightarrow B$ and $A \wedge \neg Ab_2 \Rightarrow B$.

$A \wedge \neg Ab_1 \Rightarrow B$	$A \wedge C \wedge \neg Ab_1 \Rightarrow B$	$A \wedge Ab_1 \Rightarrow B$	$A \wedge Ab_1 \Rightarrow B$	$A \Rightarrow Ab_1$	$C \Rightarrow \neg Ab_1$
$A \wedge \neg Ab_2 \Rightarrow B$	$A \wedge \neg Ab_2 \Rightarrow B$	$A \wedge Ab_2 \Rightarrow B$	$A \wedge \neg Ab_2 \Rightarrow B$	$B \Rightarrow Ab_1$	$D \Rightarrow \neg Ab_1$
(1)	(2)	(3)	(4)	(5)	(6)

Fig. 1: Ab_i-formulas in interaction

do coexist *in a same* KB_i. In this specific case, we shall assume that the knowledge engineer has thus expressed two different *sufficient* conditions for *B* to be derived from *A*. Accordingly, we shall not merge them. If his (her) goal was to express two necessary conditions, he (she) would have had to express $A \wedge \neg Ab_1 \wedge \neg Ab_2 \Rightarrow B$ directly.

Due to the specific role for Ab_i propositions, we shall not merge other types of formulas. Let us illustrate this through the main other possible forms of Ab_i-formulas in interaction.

First, it is not acceptable to merge in the same way formulas whose antecedents differ on other literals than abnormality ones, like in Table 1.(2). We shall thus not merge $A \wedge \neg Ab_1 \Rightarrow B$ and $A \wedge C \wedge \neg Ab_2 \Rightarrow B$.

To illustrate this, let us assume that the first formula asserts that when the switch is on then the lights should be on and that the second one asserts that when the switch is on and the safety fuse is ok then the lights should be on. Assume also that safety fuses are only available for a very specific trademark. Then, merging both formulas into $A \wedge C \wedge \neg Ab_1 \wedge \neg Ab_2 \Rightarrow B$ would only allow us to infer that lights should be on when this specific kind of switches is under consideration.

Formulas of the form depicted in Table 1.(3) express faulty conditions that allow us to derive B from A. Merging them into $A \wedge Ab_1 \wedge Ab_2 \Rightarrow B$ would require the *simultaneous* occurrence of faulty situations related to Ab_1 and Ab_2, respectively, in order to infer B from A. Clearly, this would be an unacceptable weakening of the faulty situations described in the initial formulas. A same analysis can be held w.r.t. to couples of formulas of the form ($Ab_1 \Rightarrow B$, $Ab_2 \Rightarrow B$).

In the general case, it does neither seem acceptable to merge formulas of the form depicted in Table 1.(4) without knowing the actual priority or specificity relation linking Ab_1 and Ab_2. For the sake of the generality, we shall neither merge them.

Since it has been assumed that each KB_i uses its own subset of Ab_i propositions and that these subsets share an empty intersection, formulas of Table 1.(5) and 1.(6) can only occur in a same KB_i. Formulas of the form of Table 1.(5) generally express *sufficient* conditions for a faulty situation to occur. In the general case, we shall not merge them so that they would just appear as necessary conditions for a faulty situation. For example, we shall not strengthen $Switch\_ko \Rightarrow Ab_1$ and $Lamp\_bulb\_ko \Rightarrow Ab_1$ into $(Switch\_ko \wedge Lamp\_bulb\_ko) \Rightarrow Ab_1$. Formulas of the form of Table 1.(6) describe conditions for not having a faulty behavior. Since in the general case, we cannot list such conditions exhaustively, they could appear as (part of) necessary conditions for not having a troublesome situation. Since we assume that they occur in a same KB_i to be merged, we however assume that the knowledge engineer has decided to give them a status of sufficient condition (otherwise, he (she) would have had to merge them into $(C \wedge D) \Rightarrow \neg Ab_1$). Finally, we shall neither merge couples of formulas taken from different categories, ranging from (1) to (6).

3.4. Merging formulas: definition and complexity

Let us generalize and formalize the above fusion schema of formulas in such a way that the syntactical form of the formulas does not matter. Accordingly, we consider KB_i under their CNF format, although we shall use non-clausal forms in many examples for the clarity of the presentation. The above treatment of necessary conditions in interaction is as follows.

Definition 3.4.1.

Let $\cup KB_i$ be the set-theoretic union of all KB_i, expressed in CNF.

Two clauses f and g are *different* iff their sets of literals they contain are different.

Definition 3.4.2.

A clause f about $f1$ is *candidate* for fusion iff

1. f is of the form $f1 \vee abnormal$, where $f1$ represents a non-empty clause formed from literals built from **P** \ **Ab** and where *abnormal* is of the form $(\vee_i Ab_i)$, and
2. f occurs in $\cup KB_i$, and
3. $\forall j$ s.t. $f \in KB_j$: $\exists f1 \vee abnormal' \in KB_j$, where *abnormal'* is of the form $(\vee_i Ab_i)$ and is different from *abnormal*.

Let us stress that under this definition, when both formulas $A \wedge \neg Ab_1 \Rightarrow B$ and $A \wedge \neg Ab_2 \Rightarrow B$ coexist in a same KB_i, they are not candidate for fusion, even if a third formula $A \wedge \neg Ab_3 \Rightarrow B$ occurs in another KB_j. Also note that this definition allows formulas of the form $A \Rightarrow Ab_1$ and $A \Rightarrow Ab_2$ to be merged into $A \Rightarrow (Ab_1 \vee Ab_2)$. For example, this would translate that merging the piece of information "When the alarm is on then there is an electrical problem" in one KB_j with "When the alarm is on then there is an hydraulical problem" in another KB_j does not require both problems to occur simultaneously when the alarm is on.

We call $\cup^+KB_i$ the set of formulas obtained from $\cup KB_i$ by performing the transformation schema of formulas described informally in the previous section. According to the above analysis, we shall only merge families of clauses of the form $f1 \vee (\vee_i Ab_i)$ when they are candidate for fusion.

Definition 3.4.3.

$Candidate\_about(f1) = \{ f$ s.t. f is of the form $f1 \vee (\vee_i Ab_i)$ and f is candidate for fusion$\}$

$Abnormal\_about(f1) = \{abnormal$ s.t. $\exists f \in Candidate\_about(f1)$ s.t. f is of the form $f1 \vee abnormal\}$

$Merged\_clause\_about(f1)$ is the clause f' of the form $f1 \vee \vee_{Abnormal\_about(f1)} (abnormal)$

$\cup^+KB_i = \cup KB_i \setminus \cup_{f1} Candidate\_about(f1) \cup \cup_{f1}\{ Merged\_clause\_about(f1)\}$

3.5. Properties of the $\cup^+$ Operator

Let us investigate the main properties of the $\cup^+$ operator.

Interestingly enough, computing $\cup^+KB_i$ is an easy task from a computational point of view

Theorem 3.5.1.

Let n be the number of clauses in $\cup KB_i$. Computing $\cup^+KB_i$ is in $O(n)$ when clauses are sorted w.r.t. a lexicographic order. It is in $O(n \log n)$ in the general case.

To better understand the $\cup^+$ operator, let us turn to its semantical properties. In particular, this will show us how this operator yields sometimes a consistent knowledge base from inconsistent ones.

Intuitively, the $\cup^+$ operator increases the set of models of $\cup KB_i$. To illustrate this, let us consider $KB_1 = \{A \wedge \neg Ab_1 \Rightarrow B\}$ and $KB_2 = \{A \wedge \neg Ab_2 \Rightarrow B\}$. $\cup^+KB_i = \{A \wedge \neg Ab_1 \wedge \neg Ab_2 \Rightarrow B\}$, whereas $\cup KB_i = \{A \wedge \neg Ab_1 \Rightarrow B, A \wedge \neg Ab_2 \Rightarrow B\}$. It is easy to check that the only interpretations that are not models of $KB_1 \cup KB_2$ contain both $\{A, \neg B\}$ and one of the three sets $\{\neg Ab_1, Ab_2\}$, $\{Ab_1, \neg Ab_2\}$ and $\{\neg Ab_1, \neg Ab_2\}$. Now, the only interpretation that is not a model of $\cup^+KB_i$ is $\{A, \neg B, \neg Ab_1, \neg Ab_2\}$. Let us characterize this more formally.

By definition, a model of $\cup^+KB_i$ that is not a model of $\cup KB_i$ must falsify at least one formula in $\cup KB_i$. Clearly, such a falsified formula in $\cup KB_i$ must be a formula that is going to be merged in $\cup^+KB_i$ since any other formula in $\cup KB_i$ belongs to $\cup^+KB_i$.

Lemma 3.5.1.

Let $\omega \in \Omega$ s.t. $\omega \in [[\cup^+KB_i]]$ and $\omega \notin [[\cup KB_i]]$.

Then $\exists f' \in \cup KB_i$ s.t. $f' \notin \cup^+KB_i$ and $\omega(f') = false$. Moreover, f' is of the form $f1 \vee Ab_l \vee \ldots \vee Ab_t$ and $\exists f \in \cup^+KB_i \setminus \cup KB_i$ where f is of the form $f1 \vee Ab_r \vee \ldots \vee Ab_s$, where $\{ Ab_l,\ldots, Ab_t\} \subset \{ Ab_r ,\ldots, Ab_s\}$.

Accordingly, let us define the set of additional models as follows:

Definition 3.5.1.

Additional_Models($\cup^{+}KB_i$) is the set of all interpretations $\omega \in \Omega$ s.t.

1. $\exists f \in \cup^{+}KB_i \setminus \cup KB_i$ where f is of the form $f1 \vee Ab_r \vee \ldots \vee Ab_s$ s.t. $\omega \supset \{\neg f1, \{\neg Ab_i,\ldots, \neg Ab_j\}\}$, where

   (*)$\{\neg Ab_i,\ldots, \neg Ab_j\} \subset \{\neg Ab_r,\ldots, \neg Ab_s\}$

   (**)$\exists f' \in \cup KB_i$ of the form $f1 \vee Ab_l \vee \ldots \vee Ab_t$ s.t. $\{\neg Ab_l,\ldots, \neg Ab_t\} \subset$

   $\{\neg Ab_i,\ldots, \neg Ab_j\}$

2. $\omega \in [[\ \cup^{+}KB_i\ ]]$

Condition 1 expresses that ω satisfies one merged formula f while it falsifies at least one formula f' from $\cup KB_i$ that is going to be merged. Condition 2 ensures that ω is a model of $\cup^{+}KB_i$. In the general case, it is difficult to drop Condition 2 and refine Condition 1 without knowing the actual contents of every KB_i. For example, it would be tempting to drop Condition 2 and strengthen Condition 1, with $\exists\, Ab_k \in \{Ab_r,\ldots, Ab_s\}$ s.t. $\neg Ab_k \notin \omega$. Indeed, although such a condition ensures that the resulting merged formula is satisfied by ω, other formulas in $\cup^{+}KB_i$ can be falsified. For example, let $KB_1 = \{f1 \vee Ab_1\}$ and $KB_2 = \{f1 \vee Ab_2, f2, f2 \Rightarrow Ab_1\}$. Then $\omega = \{f2, \neg f1, Ab_1, \neg Ab_2\}$ satisfies the strengthened Condition 1. However, it contradicts the formula $f2 \Rightarrow Ab_1 \in \cup^{+}KB_i$ and is thus not a model of $\cup^{+}KB_i$.

Let us now show that *Additional_Models* exactly describes the additional models introduced by the merging operator $\cup^{+}$.

Theorem 3.5.2.

1. *Additional_Models*($\cup^{+}KB_i$) $\cap\ [[\ \cup KB_i\ ]] = \emptyset$

2. $[[\cup^{+}KB_i\ ]] = [[\ \cup KB_i\ ]] \cup$ *Additional_Models*($\cup^{+}KB_i$)

Interestingly enough, the $\cup^{+}$ operator allows one to restore consistency in some circumstances. Indeed,

Theorem 3.5.3.

(a) $[[\cup^{+}KB_i\ ]] = \emptyset \Rightarrow [[\ \cup KB_i\ ]] = \emptyset$

(b) $[[\ \cup KB_i\ ]] = \emptyset \not=> [[\cup^{+}KB_i\ ]] = \emptyset$

4. PRECISE INFORMATION CAN BE HIDDEN BY A MORE GENERAL ONE

When several information sources are to be fused, another important drawback of logic-based approaches is that weaker information is hidden in deduction and in most logic-based inference processes. Indeed, more precise information is subsumed by more general one from a deductive point of view in the sense that the extra information that it conveys can be hidden by the more general one. We believe that this problem is a crucial one that is too much neglected when logic-based information fusion is considered.

For example, assume that one source asserts the piece of information "*When the switch is on then the light are on*" and the second one asserts the more precise one "*When the switch is on and the fuse is ok then the light are on*". Taking both pieces of information into account, a deductive agent will be able to infer that the lights are on when the switch is on, even when it knows that the fuse is not ok! Now, the problem is that forms of subsumption of more specific information are not always as apparent as in the above example. Actually, it might concern any logical consequence of any of the information sources. With the possible exception of linear logic, there is no general satisfactory logical tool that can avoid by itself this drawback. Accordingly, we discuss a possible methodology based on the restricted use of a controlled logical language and on a-priori tests that prevents this drawback from occurring.

Let us for instance consider $KB_1 = \{A \Rightarrow B, A \wedge \neg Ab_1 \Rightarrow B\}$. Clearly, $A \Rightarrow B$ subsumes $A \wedge \neg Ab_1 \Rightarrow B$ and $\neg Ab_1$ is thus not required to be *true* for B to be inferred from A. Actually, detecting such situations might require us to consider the whole contents of the knowledge bases. As an example, we can see that $KB_1 = \{A \Rightarrow C, C \Rightarrow B, A \wedge \neg Ab_1 \Rightarrow B\}$ leads to an identical problem.

Actually, we shall assume that each designer of a KB_i has checked that his (her) knowledge base KB_i satisfies the following assumptions that prevent the condition for the *absence of faulty situation* from being overridden.

Assumption 4.1. *Conditions for the absence of faulty situations are not overridden* (1).

Let $f1$ represent a clause formed from **P** \ **Ab**.

$\forall f \in KB_i$ s.t. f is of the form $f1 \vee (\vee_i Ab_i)$ we have either $[[KB_i]] = \emptyset$ or $KB_i \not\models f1$.

Clearly, the above examples in this section do not satisfy this assumption. Indeed, the CNF version of $A \wedge \neg Ab_1 \Rightarrow B$ being $\neg A \vee B \vee Ab_1$, we have both $[[KB_i]] \neq \emptyset$ and $KB_i \models \neg A \vee B$ (which is equivalent to $KB_i \models A \Rightarrow B$).

Let us propose an even stronger assumption that states that a similar constraint holds even when a failure can be consistently assumed.

Assumption 4.2. *Conditions for the absence of faulty situations are not overridden (2).*

Let $f1$ represent a clause formed from **P** \ **Ab**.

$\forall f \in KB_i$ s.t. f is of the form $f1 \vee (\vee_i Ab_i)$ we have either $[[KB_i \wedge (\vee_i Ab_i)]] = \emptyset$ or $KB_i \wedge (\vee_i Ab_i) \not\models f1$.

Once again, the above examples in this section do not satisfy this assumption.

Properties 4.1. Clearly, assumption 4.2 is stronger than assumption 4.1, at least in the most expected usual cases, namely when both $[[KB_i]] \neq \emptyset$ and $[[KB_i \wedge (\vee_i Ab_i)]] \neq \emptyset$ (the last condition expressing that a failure can be consistently assumed). In this case $KB_i \wedge (\vee_i Ab_i) \not\models f1$ entails $KB_i \not\models f1$, while the converse is not valid in the general case.

It is easy to derive more focused assumptions taking the specific structures of clauses into account. Let us express a natural one that concerns implication formulas.

Assumption 4.3. *Conditions for the absence of faulty situations are not overridden (3).*

Let g, h be any formulas built from **P** \ **Ab**.

$\forall f \in KB_i$ s.t. f is of the form $g \wedge (\wedge_i \neg Ab_i) \Rightarrow h$

when $KB_i \not\models \wedge_i \neg Ab_i$ and $KB_i \not\models \neg g$ we have that $KB_i \wedge (\vee_i Ab_i) \not\models g \Rightarrow h$

It is straightforward to see that the above examples in this section do neither satisfy this assumption. Similar assumptions should be tested against $\cup^{+}KB_{\mathrm{i}}$ as well, due to the interaction of the various knowledge bases, as the following example illustrates it: $KB_1 = \{A \Rightarrow C, A \wedge \neg Ab_1 \Rightarrow B\}$ and $KB_2 = \{C \Rightarrow B\}$. Let us just state the assumption corresponding to the latter defined one.

Assumption 4.4. *Conditions for the absence of faulty situations are not overridden (4).*

Let g, h be any formula built from **P** \ **Ab**.

$\forall f \in \cup^{+}KB_{\mathrm{i}}$ s.t. $f = g \wedge (\wedge_{\mathrm{i}} \neg Ab_{\mathrm{i}}) \Rightarrow h$

when $\cup^{+}KB_{\mathrm{i}} \not\models \wedge_{\mathrm{i}} \neg Ab_{\mathrm{i}}$ and $\cup^{+}KB_{\mathrm{i}} \not\models \neg g$ we have that $\cup^{+}KB_{\mathrm{i}} \wedge (\vee_{\mathrm{i}} Ab_{\mathrm{i}}) \not\models g \Rightarrow h$

Let us conclude this section by two remarks.

Although computing any of the above assumptions w.r.t. a given clause f is co-NP-complete, it should not prevent the knowledge engineer from performing it often efficiently, as recent progress in propositional deduction and search have shown it.

In the general case, it does not seem relevant to propose subsumption tests for clauses containing negative occurrences of abnormality propositions. For example, $A \wedge Ab_1 \Rightarrow B$ ($\neg A \vee \neg Ab_1 \vee B$ in CNF) is subsumed by $A \Rightarrow B$. It translates that *even* under a faulty condition of the device, B can be inferred from A. Although, it could be useful for the knowledge engineer to know that the rule $A \Rightarrow B$ already asserts it unconditionally, there is no reason to forbid it.

5. CONCLUSIONS

In this paper, three currently hot research topics about knowledge fusion in a logical framework have been presented. They are related to three possibly unwanted abilities of ideally rational agents. The first one is the need to reason in a consistent way in the presence of contradictions. The second one is linked to the possible weakening of information when submitted to a fusion process. The third one is due to the fact that more general pieces of information can hide more specific and more precise ones. For each of these problems, motivating examples have been introduced and solutions investigated.

6. ACKNOWLEDGMENTS

This work has been supported by the IRCICA Project "*Composants de connaissance intelligents*".

7. REFERENCES

[1] C. Baral, S. Kraus, J. Minker and V.S. Subrahmanian, «Combining knowledge bases consisting of first-order theories», *Computational Intelligence*, **8**(1), pp. 45-71, 1992.

[2] S. Benferhat, D. Dubois, S. Kaci and H. Prade, « Encoding classical fusion in possibilitic logic: a general framework for rational syntactic merging », *Proc. of ECAI'2000*, pp. 8-12, 2000.

[3] I. Bloch and A. Hunter (guest eds), "Fusion: general concepts and characteristics", *Int. Journ. of Intelligent Systems*, vol. 16, 2000.

[4] L. Cholvy, «Reasoning about merging information », *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, **3**, pp. 233-263, 1998.

[5] M. Dalal, « Investigations into a theory of knowledge-base revision: preliminary report », *Proc. AAAI'88*, pp. 475-479, 1988.

[6] É. Grégoire, « Fusion under the closed-world assumption », *Proc. of Sensor Fusion : Architectures, Algorithms, and applications 5*, Belur V. Dasarathy (ed.), pp. 197-204, 2001.

[7] É. Grégoire and S. Konieczny, « Logic-based information fusion in artificial intelligence », *Information Fusion*, Elsevier, vol. 7, n° 1, 2005.

[8] É. Grégoire, « An unbiased approach to iterated fusion by weakening », *Information Fusion*, Elsevier, vol. 7, n° 1, 2005.

[9] É. Grégoire and S. Konieczny (guest editors), Logic-based Information Fusion: special issue, *Information Fusion*, Elsevier, 2006.

[10] É. Grégoire, « Fusing Cooperative Technical-Specification Knowledge Components », *Int. Journ. on Artificial Intelligence Tools (IJAIT),* vol. 12, n° 3, pp. 265-278, September 2003.

[11] É. Grégoire, « About the dynamics of iterated knowledge fusion by weakening », *Proc. of the 2005 IEEE Int. Conf. on Information Reuse and Integration (IEEE-IRI'05)*, pp. 326-331, Las Vegas, August 2005.

[12] É. Grégoire, « About the use of semaphores to handle contradictory knowledge sources », *Proc. of the 15th IEEE Int. Conf. on Tools with Artificial Intelligence (ICTAI'2003)*, pp. 213-218, Sacramento, November 2003.

[13] É. Grégoire, « Extension of a distance-based fusion framework », *Proc. of the 8th Int. Conf. on Sensor Fusion : Architectures, Algorithms and Applications,* Belur V. Dasarathy (ed), pp. 282-286, Orlando, April 2004.

[14] W. Hanscher, L. Console and J. de Kleer, *Readings in model-based reasoning*, Morgan Kaufmann, 1992.

[15] J. Lin and A.O. Mendelson, «Merging databases under constraints », *Int. Journ. of Cooperative Information Systems*, **7**(1), pp. 55-76, 1998.

[16] J. McCarthy, « Applications of circumscription to formalizing common-sense reasoning », *Art. Int.*, **29**, pp. 89-116, 1986.

[17] R. Reiter, «A theory of diagnosis from first principles », *Art. Int.,* **32**, pp. 81-132, 1987.

[18] N. Rescher and R. Manor, « On inference from inconsistent premises », *Theory and Decision*, **1**, pp. 179-219, 1970.

[19] P.Z. Revesz, « On the semantics of theory change: arbitration between old and new information », Proc. of the 12th ACM SIGACT-*SIGMOD-SIGART Symp. on Principles of Databases*, pp. 71-92, 1993.

[20] P.Z. Revesz, « On the semantics of arbitration », *Int. Journ. of Algebra and Computation*, **7**(2), pp. 133-160, 1997.

[21] V.S. Subrahmanian, «Amalgamating knowledge bases », *ACM Transactions on Database Systems*, **19**(2), pp. 291-331, 1994.

Logical traps in high-level knowledge and information fusion

E. Grégoire

CRIL CNRS & IRCICA

Lens - France

What is the paper about?

- Hot research issues about the fusion of symbolic knowlege expressed in logic
- 3 main problems are motivated and promising solutions are presented

What is the paper about?

Three main issues

- Dramatic effects of *logical inconsistency*
- Necessary conditions can become sufficient ones
- Precise information can be hidden by more general knowledge

What is the paper about?

Framework

- Symbolic information and knowledge expressed in logic

- Full deductive reasoning capacities

1. Logical inconsistency

- From any contradictory pieces of information, a sound and complete logical reasoner can infer anything (and its contrary)

- A main issue in logic-based fusion is how contradictory information should be handled

1. Logical inconsistency

Example

- $KB_1 \supset \{\mathit{sensor1\_ok}\}$
- $KB_2 \supset \{\text{not}(\mathit{sensor1\_ok})\}$

- $KB_1 \cup KB_2 \models f \quad (\forall f)$
- $KB_1 \cup KB_2$?$\models$ *Launch-missile* $\Rightarrow$ Yes
- $KB_1 \cup KB_2$?$\models$ *not(Launch-missile)* $\Rightarrow$ Yes

1. Logical inconsistency

How to solve such a problem?

- *Weaken the inference systems*
  - *paraconsistent logics*
  - *incomplete reasoners*
  - *reason from maximal consistent subsets of formulas*

1. Logical inconsistency

How to solve such a problem?

- *Detect inconsistencies*
  - A heavy task!
  - In the basic Boolean clausal framework, it is NP-complete

2. Necessary conditions can become sufficient ones

Example

$KB_1 \supset$ {If a target is detected and *<u>if it is not out of reach</u>* then open fire against the target}

$KB_2 \supset$ {If a target is detected and *<u>if there is not a more seriously threatening target</u>* then open fire against the target}

….two necessary conditions for opening fire against the target

2. Necessary conditions can become sufficient ones

$KB_1 \cup KB_2$ will contain both rules

- $KB_1 \cup KB_2 \cup$ {*Target detected, Target not out of reach, There is a more seriouly threatening target*}   ? |= open fire at the target

  $\Rightarrow$ *Yes*   *which is a wrong decision...*

In this simple case, the problem is due to the fact that the rules themselves should have been fused into

If a target is detected and *<u>if it is not out of reach</u>* and *<u>if there is not a more seriously threatening target</u>* then open fire against the target

2. Necessary conditions can become sufficient ones

Actually finding all forms of such unexpected behaviors is not that easy

A case study is presented in the paper, in a non-monotonic Boolean framework using abnormality conditions

2. Necessary conditions can become sufficient ones

When the switch is on and when the switch is not out of service, then the lights are on

$$Switch\_on \wedge \neg Ab_1 \Rightarrow Lights\_on$$

$$Out\_of\_service\_switch \Rightarrow Ab_1$$

In the general case, we have that

$A \wedge \neg Ab_1 \Rightarrow B$ and $A \wedge \neg Ab_2 \Rightarrow B$

are merged to form $A \wedge \neg Ab_1 \wedge \neg Ab_2 \Rightarrow B$

2. Necessary conditions can become sufficient ones

But many cases must be considered!

$A \wedge \neg Ab_1 \Rightarrow B$ $A \wedge C \wedge \neg Ab_1 \Rightarrow B$ $A \wedge Ab_1 \Rightarrow B$ $A \wedge Ab_1 \Rightarrow B$ $A \Rightarrow Ab_1$ $C \Rightarrow \neg Ab_1$

$A \wedge \neg Ab_2 \Rightarrow B$ $A \wedge \neg Ab_2 \Rightarrow B$ $A \wedge Ab_2 \Rightarrow B$ $A \wedge \neg Ab_2 \Rightarrow B$ $B \Rightarrow Ab_1$ $D \Rightarrow \neg Ab_1$

2. Necessary conditions can become sufficient ones

Let $\cup KB_i$ be the set-theoretic union of all KB_i, expressed in CNF.

Two clauses *f* and *g* are *different* iff their sets of literals they contain are different.

2. Necessary conditions can become sufficient ones

A clause *f* about *f1* is *candidate* for fusion *iff*

- *f* is of the form $f1 \vee abnormal$, where *f1* represents a non-empty clause formed from literals built from **P** \ **Ab** and where *abnormal* is of the form $(\vee_i Ab_i)$, and
- .*f* occurs in $\cup KB_i$, and
- $\forall j$ s.t. $f \in KB_j : \exists f1 \vee abnormal' \in KB_j$, where *abnormal'* is of the form $(\vee_i Ab_i)$ and is different from *abnormal*.

2. Necessary conditions can become sufficient ones

$Candidate\_about(f1) = \{ f$ s.t. f is of the form $f1 \vee (\vee_i Ab_i)$ and

f is candidate for fusion$\}$

$Abnormal\_about(f1) = \{abnormal$ s.t. $\exists f \in Candidate\_about(f1)$ s.t.

f is of the form $f1 \vee abnormal\}$

$Merged\_clause\_about(f1)$ is the clause f' of the form

$$f1 \vee \vee_{Abnormal\_about(f1)} (abnormal)$$

$$\cup^+ KB_i = \cup KB_i \setminus \cup_{f1} Candidate\_about(f1) \cup \cup_{f1} \{ Merged\_clause\_about(f1) \}$$

2. Necessary conditions can become sufficient ones

Computational results

Let n be the number of clauses in $\cup KB_i$.

Computing $\cup^{+}KB_i$ is in $O(n)$ when clauses are sorted w.r.t. a lexicographic order.

It is in $O(n \log n)$ in the general case.

=> a fast pre-processing step can solve the problem !

3. More precise information can be hidden

Example

$KB_1 \supset$ {*When the switch is on then the lights are on*}

$KB_2 \supset$ {*When the switch is on and the fuse is ok then the lights are on*}

$KB_1 \cup KB_2 \cup$ {*the switch is on, the fuse is not ok*} ?|= lights are on

$\Rightarrow$ Yes

But this problem is not always that apparent...

3. More precise information can be hidden

$KB_1 = \{A \Rightarrow B, A \wedge \neg Ab_1 \Rightarrow B\}$

Actually, detecting such situations might require us to consider the whole contents of the knowledge sources

$KB_2 = \{A \Rightarrow C, C \Rightarrow B, A \wedge \neg Ab_1 \Rightarrow B\}$ leads to an identical problem.

3. More precise information can be hidden

- We must ensure that each designer of a KB_i has checked that his (her) KB_i satisfies the following assumptions that prevent the condition for the *absence of faulty situation* from being overridden.

Assumption 1. *Conditions for the absence of faulty situations are not overridden* (1).

- Let *f1* represent a clause formed from **P** \ **Ab**.

- $\forall f \in KB_i$ s.t. f is of the form $f1 \vee (\vee_i Ab_i)$ we have either $[[KB_i]] = \emptyset$ or $KB_i \not\models f1$.

3. More precise information can be hidden

An even stronger assumption that states that a similar constraint holds even when a failure can be consistently assumed.

Assumption 2. *Conditions for the absence of faulty situations are not overridden (2).*

- Let $f1$ represent a clause formed from **P** \ **Ab**.
- $\forall f \in KB_i$ s.t. f is of the form $f1 \vee (\vee_i Ab_i)$ we have either $[[KB_i \wedge (\vee_i Ab_i)]] = \emptyset$ or $KB_i \wedge (\vee_i Ab_i) \not\models f1$

3. More precise information can be hidden

The specific structures of clauses can be taken into account. For implication formulas:

Assumption 3. *Conditions for the absence of faulty situations are not overridden (3).*

- Let g, h be any formulas built from **P \ Ab**.
- $\forall f \in KB_i$ s.t. f is of the form $g \wedge (\wedge_i \neg Ab_i) \Rightarrow h$
- when $KB_i \not\models \wedge_i \neg Ab_i$ and $KB_i \not\models \neg g$ we have that $KB_i \wedge (\vee_i Ab_i) \not\models g \Rightarrow h$

3. More precise information can be hidden

Similar assumptions should be tested against $\cup^{+}KB_{i}$ as well, due to the interaction of the various knowledge sources

Example:

$KB_1 = \{A \Rightarrow C, A \wedge \neg Ab_1 \Rightarrow B\}$ and $KB_2 = \{C \Rightarrow B\}$.

Conclusions

Fusing symbolic information is not that straightforward !

Unexpected conclusions can be drawn using basic deductive reasoning based on *directly* fused knowledge

In the paper, we have focused on three such issues. For each of them, we have emphasised their importance, and sketched some possible practical solutions.